Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчёт**

**по лабораторной работе №9**

**Дисциплина: ОБРАБОТКА БОЛЬШИХ ДАННЫХ**

**Тема: «ЭЛЕМЕНТАРНЫЕ СПОСОБЫ РАБОТЫ С ГРАФАМИ В R»**

Работу выполнила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.В.Стасюк

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Математическое и программное обеспечение

компьютерных технологий

Преподаватель

ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.И.Шиян

**Цель работы:** научиться работать с графами. Освоить основные функции обработки графов – их задание, визуализация, оформление, извлечение информации о графе.

**Задание:**

1. Создайте кольцевой граф g со случайным числом вершин G\_size (от N+10 до (N/10+5)2+5N). Выведите число ребер и вершин этого графа. Постройте граф, выведите его матрицу смежности.

2. Создайте граф g1 из пустого графа с числом вершин G\_size желтого цвета. Добавьте ему N\*8 случайных ребер, сформированных из вектора вершин, окрасьте ребра красным цветом, нарисуйте граф и выведите его матрицу смежности. Добавьте графу g1 еще N\*10 случайных ребер, сформированных из вектора вершин, окрасьте ребра синим цветом, нарисуйте граф и выведите его матрицу смежности.

3. Добавьте ребра между вершиной 2N+23 и 2N+20, 2N+12 и N+15, 2N‑1 и N+8, 2N и 2N+1, N+7 и N+13, окрасьте их в черный цвет (предварительно проверьте существуют ли такие вершины – функцией %in% либо match, для несуществующих вершин ребра не добавляйте). Нарисуйте граф. Выведите соседей N-й вершины, ребра, инцидентные этой вершине. Соединены ли вершины N+10 и N+12? Выведите матрицу смежности.

4. Добавьте еще одну вершину и подключите ее к той, которая имеет наибольшее количество связанных с ней узлов. Присвойте имена всем вершинам (например, буквы в алфавитном порядке – используйте заглавные и, если не хватит, строчные буквы). Выведите матрицу смежности. Выберите вершины, для которых значение связности меньше 5 и больше 2.

5. Испробуйте алгоритмы размещения Вашего графа (in\_circle, in\_tree, lattice). Результаты включить в отчет.

6. Выполните измерение диаметра графа g1, выведите список самых коротких путей для каждой вершины и откалибруйте величины вершин согласно их степеням.

7. Имеется M (M < 50) городов, в каждом из которых открылась кооперативная парикмахерская. Известна стоимость стрижки в каждом городе и стоимость проезда между городами. Известно, что не между всеми городами есть прямая дорога. Стоимость и проезд выражаются неотрицательными вещественными числами. Для агентов каждого города определить город, куда им следует съездить, чтобы подстричься самым дешевым образом (в своем собственном городе из соображений секретности стричься нельзя). Данные сформировать случайным образом.

**Ход работы:**

1. Создадим кольцевой граф g со случайным числом вершин G\_size (от N+10 до (N/10+5)2+5N). Выведем число ребер и вершин этого графа (рисунок 1). Построим граф (рисунок 2), выведем его матрицу смежности (рисунок 3).

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 1 – Число ребер и вершин графа g

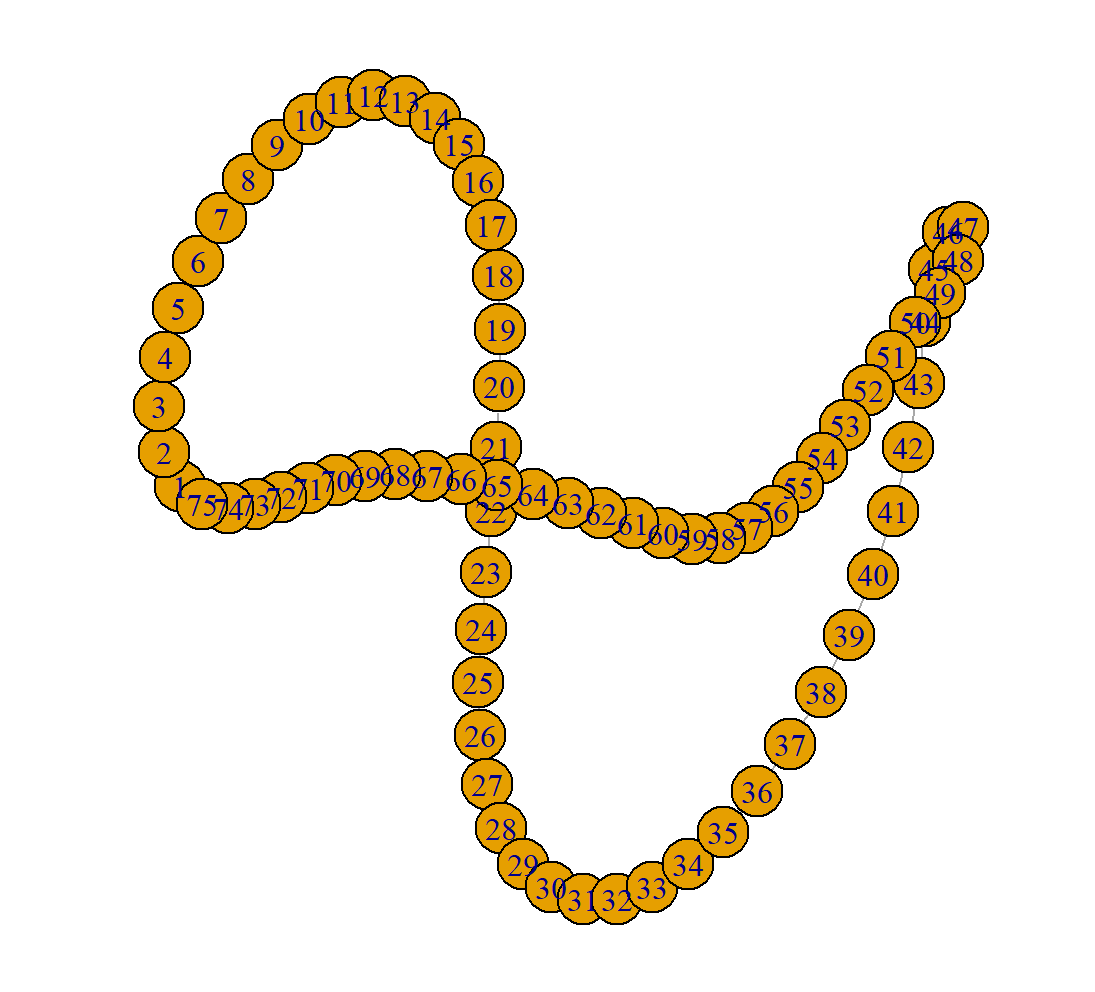


Рисунок 2 – Кольцевой граф g

Изображение выглядит как текст, снимок экрана, число, линия

Автоматически созданное описание

Рисунок 3 – Матрица смежности графа g

1. Создадим граф g1 из пустого графа с числом вершин G\_size желтого цвета. Добавим ему N\*8 случайных ребер, сформированных из вектора вершин, окрасим ребра красным цветом, нарисуем граф (рисунок 4). Добавим графу g1 еще N\*10 случайных ребер, сформированных из вектора вершин, окрасим ребра синим цветом, нарисуем граф (рисунок 5).

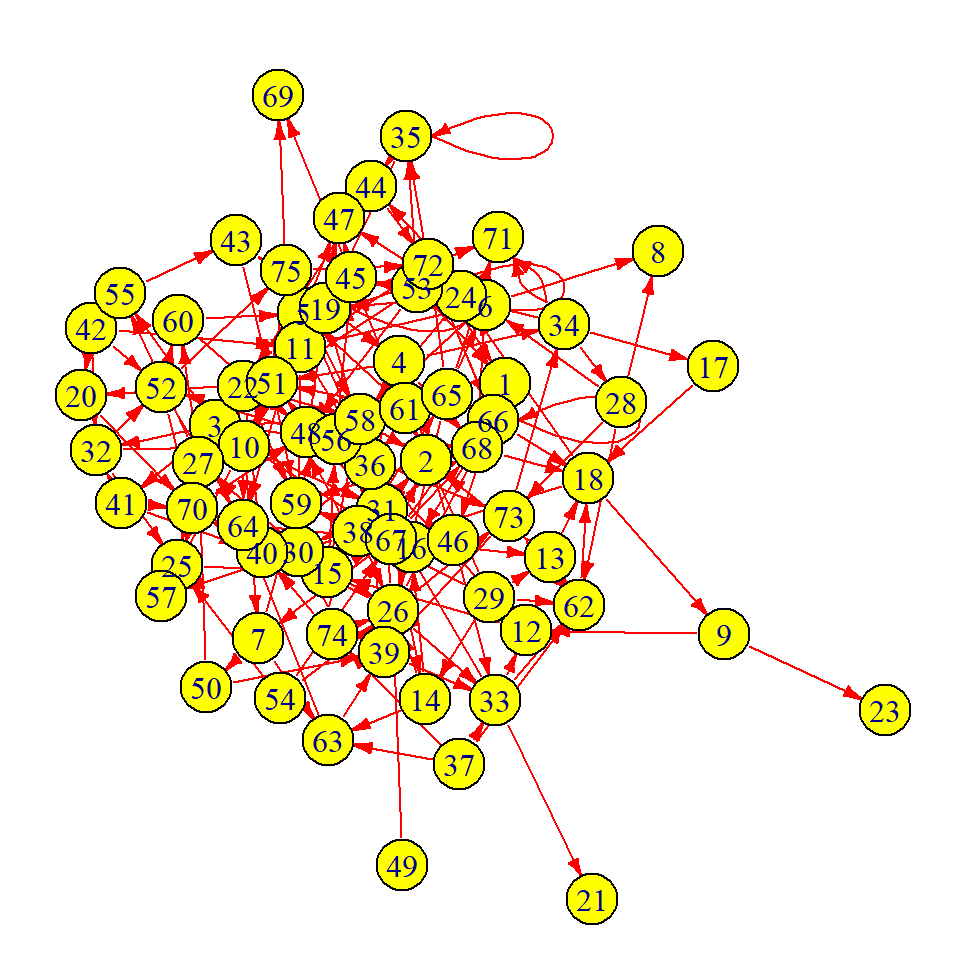


Рисунок 4 – Граф первого варианта

Изображение выглядит как Симметрия, Красочность, круг, искусство

Автоматически созданное описание

Рисунок 5– Граф второго варианта

3. Добавим ребра между вершиной 2N+23 и 2N+20, 2N+12 и N+15, 2N-1 и N+8, 2N и 2N+1, N+7 и N+13, окрасим их в черный цвет. Нарисуем граф (рисунок 6). Выведем соседей N-й вершины (рисунок 7), ребра,

инцидентные этой вершине (рисунок 8). Выясним, соединены ли вершины N+10 и N+12 (рисунок 9)

Изображение выглядит как Графика, искусство

Автоматически созданное описание

Рисунок 6– Граф с чёрными рёбрами

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 7 – Соседи N-й вершины

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание

Рисунок 8 – Ребра, инцидентные N-ой вершине

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Рисунок 9 – Проверка на соединение вершин N+10 и N+12

4. Добавим еще одну вершину и подключим ее к той, которая имеет наибольшее количество связанных с ней узлов. Присвоим имена всем вершинам (например, буквы в алфавитном порядке – используем заглавные и, если не хватит, строчные буквы). Построим граф (рисунок 10). Выберем вершины, для которых значение связности меньше 5 и больше 2 (рисунок 11).

Изображение выглядит как Красочность

Автоматически созданное описание

Рисунок 10 – Граф с новой вершиной

Изображение выглядит как текст, Шрифт, рукописный текст, линия

Автоматически созданное описание

Рисунок 11 – Вершины, значение связности меньше 5 и больше 2

5. Испробуем алгоритмы размещения графа (in\_circle (рисунок 12), in\_tree (рисунок 13)).

Изображение выглядит как круг, искусство, Красочность, рисунок

Автоматически созданное описание

Рисунок 12 – Размещение графа как in\_circle

Изображение выглядит как линия, Детское искусство, искусство, дизайн

Автоматически созданное описание

Рисунок 13 – Размещение графа как in\_tree

6. Выполним измерение диаметра графа g1 (рисунок 14), выведем список самых коротких путей для каждой вершины (рисунок 15) и откалибруем величины вершин согласно их степеней (рисунок 16).

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 14 – Диаметр графа

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 15 – Часть списка самых коротких путей для каждой вершины

Изображение выглядит как Красочность, круг, искусство

Автоматически созданное описание

Рисунок 16 – Откалиброванный граф

7. Первым шагом мы генерируем случайные данные, такие как стоимость стрижки в каждом городе и стоимость проезда между городами. Далее создаем граф g: граф с вершинами, представляющими города, и ребрами, представляющими стоимости проезда между городами. Ребра имеют веса, соответствующие стоимостям проезда. Каждой вершине графа присваивается атрибут "haircut\_cost", содержащий стоимость стрижки в этом городе.

Изображение выглядит как рисунок, зарисовка, Симметрия, искусство

Автоматически созданное описание

Рисунок 17 – Полученный граф

Далее находим самый дешевый город для стрижки и цену стрижки в нем

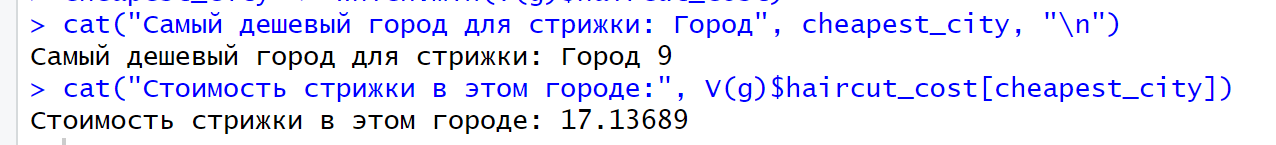


Рисунок 18 – Необходимые значения